

Best selling book on QTP

2nd Edition

QuickTest Professional Unplugged



Tarun Lalwani

Winner of 2nd ATI Automation Honors Award

QuickTest Professional

Unplugged

SECOND EDITION

By

Tarun Lalwani

QuickTest Professional Unplugged

By *Tarun Lalwani*

Editor: *Siddharth N Kanoujia*

Technical Editor: *Terry Howarth*

Reviewer: *Mark Smith and Terry Howarth*

ISBN: 978-0-9836759-1-4

Printing History:

First Edition: May 2009 (ISBN: 978-0-5780257-9-7)

Second Edition: September 2011

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise without either the prior written permission of the Author, or authorization through payment of the appropriate per-copy fee to the Author. For permission please contact author at [**KnowledgeInbox.com/contact-us**](http://KnowledgeInbox.com/contact-us).

Source codes discussed in this book can be copied, modified or distributed without permission from author or the publisher by including the below mentioned reference

‘Source code taken from “QuickTest Professional Unplugged” By Tarun Lalwani

‘The link for downloads is [**KnowledgeInbox.com/demos/QTPUnplugged_SourceCodes.zip**](http://KnowledgeInbox.com/demos/QTPUnplugged_SourceCodes.zip).

‘Website: [**KnowledgeInbox.com/books/quicktest-professional-unplugged**](http://KnowledgeInbox.com/books/quicktest-professional-unplugged)

This document also contains registered trademarks, trademarks and service marks that are owned by their respective companies or organizations. The Publisher and the author disclaim any responsibility for specifying which marks are owned by which companies or organizations.

Copyright (2008) Mercury Interactive (Israel) Ltd. Reproduced with permission

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

Contents

Preface	xiii
Who This Book Is For	xiv
What's New in the Second Edition	xiv
Source Code in the Book	xiv
Feedback and Queries	xiv
Ordering this Book	xiv
Acknowledgements	xv
Quotes From Reviewers	xv
I. Introduction (Updated)	I
What is Test Automation?	1
When Should Test Automation Be Used?	1
When Should Test Automation Be Avoided?	1
The Automated Testing Process	2
What is HP QuickTest Professional (QTP)	3
Getting Started with QTP	4
Downloading QTP from HP website	4
Installing QTP	5
QTP Licensing Model	8
Seat License	8
Floating License	9
2. Using QTP Help	13
Contents Tab	14
Properties returned by the GetROProperty Function	14
Methods provided by a Test Object	14
Index Tab	15
Search Tab	16

3. Object Repository (OR)	19
How objects are added to the OR	20
Test and Run-time Objects	22
TO Properties	22
Object Repository Modes	24
Per-Action versus Shared Object Repository	24
The Object Spy	25
Object Identification	27
User-defined Objects	28
Object Repository Shortcomings	29
4. Data Tables	33
Design and run-time data table	35
Design time data table	35
Run-time data table	35
When to use the global or a local data table	35
Setting data table iterations	36
Data table object model	38
Data table formatting	38
5. Actions	43
Input and Output Parameters	43
Types of Actions	46
Inserting Calls to Actions	46
Insert Call to New...	47
Insert Call to Existing...	48
Insert Call to Copy...	49
6. QTP Environment Variables	51
Types of Environment variables	51
Environment Object	52
7 Utility Objects	61
The Crypt Object	61
The OptionalStep Object	62
The PathFinder Object	62
The RandomNumber Object	63
The Setting Object	63
The WebUtil Object	65
The SystemUtil Object	65
The RegisterUserFunc Object	66
8. Checkpoints	69
Types of checkpoints	69
Problems with QTP built-in checkpoints	70
Alternatives to QTP checkpoints	71

Updating Checkpoints at run-time	71
9. Library Files	75
Associating a Library globally with a Test	75
Dynamically Loading libraries locally at run-time	76
Dynamically Loading libraries globally at run-time	76
Understanding Execution Scope	78
Applicability of Option Explicit	79
Executing code in Local scope from within Global Scope	79
10. Descriptive Programming (DP).	83
Object Identification.	83
Implicit Properties	84
When to use Descriptive Programming	84
Descriptive Programming Syntax	84
Using description objects.	84
The micclass Identifier.	86
Using Description Strings	86
Enumerating ChildObjects.	86
Converting an OR-based script to a DP-based script.	87
Using a VBScript DP object repository	89
Problems with Ordinal Identifiers	90
Working with Multiple Browser	92
Browser identification Issue.	93
Browser identification using OpenTitle	93
Browser Identification using a unique OpenURL Property	93
11. Debugging in QTP	97
Establishing our Debugging Configuration	97
Using Breakpoints	98
Working with the Watch Expression Tab	98
Working with the Variables Tab	99
Working with the Command Tab	100
12. Recovery Scenarios	103
When not to use recovery scenarios?	103
Situations in which a Recovery Scenario won't work	107
Recovery Scenario for IE Authentication window	109
Default Recovery Scenarios.	110
The Recovery object	111
Recovery Function Calls	111
Error Handling	113
13. Regular Expressions	117
Regular Expression Characters	118
When to use Regular Expressions in QTP.	121

Testing a Regular Expression Using Code	123
Using a Regular Expression to Select a WebList or WebRadioGroup Value	124
Extracting Information Using a Regular Expression	124
Replacing Data from a String Using a Regular Expression	125
14. VBScript	129
Strings	129
Conversion function	137
Date and Time functions	138
Misc	141
TypeName	141
VarType	141
GetRef	141
CreateObject	142
GetObject	142
Working with Arrays	148
Fixed length arrays	148
Dynamic Arrays	148
Execute Statement	149
Eval function	150
Optional arguments in VBScript	150
15. Synchronization	157
Synchronization Points	157
1st Method – Using the Sync method	157
2nd Method – Using the Wait statement	158
3rd Method – WaitProperty method	158
Checking Object Existence	158
Browser Vs Page Sync	159
Events	159
Replay Options	161
16. Test Results	165
Filtering Steps in a Report	166
Reporting Custom Steps	167
Inserting Files in Test Results	167
Inserting Snapshots in Test Results	168
Converting XML Results to HTML	170
Configuring QTP to generate HTML results	170
The QTP Reporter Object	171
Accessing Test Results at the end	173
17. Working with APIs	177
Extern object	177
VB API Definition Syntax	177

QTP API Definition	178
QTP API Limitation	182
API using COM Objects in VB6	182
Running API using Excel	185
Dynamically generating an Excel Macro	187
Working with Modal dialog boxes	190
18. QTP Automation Object Model	195
QTP Automation Object Model (AOM)	195
Passing arguments to Test Cases	196
19. HTML DOM	201
When to use the HTML DOM	201
When not to use the HTML DOM	202
HTML DOM Objects	202
HTML Node	202
The Document Object	203
The HTML Element Object	203
The HTML Element Collection	203
Getting the Web page Element using the DOM	204
Other HTML Elements	205
Converting a DOM object to QTP Test Object	208
Checking Appearance using Style Sheets	208
Checking Visibility of a Object on a Page	208
Selecting a Radio button by Text	210
20. Working with Web Tables	217
Web Tables	217
Accessing the WebTable	218
Using Index	219
Using Name or HTML ID	219
Using innerText/outerText	219
Using OR	222
Using an object inside the table	224
Getting the location of an object inside the table	225
Clicking inside a WebTable	227
Asynchronous Table loading	229
Exporting WebTable to a DataTable	229
Extending WebTable functionality	230
Getting a QTP Table from a element	230
Finding a cell in a Table	231
21. Working with XML	237
QTP XML Objects	238
Reading and Modifying XML	238

Working with Multiple environments of an Application	241
Exporting XML to a DataTable	243
Comparing XML	244
22. Working with Databases	249
Connecting to a Database.	249
Building Connection Strings	250
Checking the State of a Connection or Recordset	253
Exporting the Results of a Query to Data Table	254
Executing a Stored DB procedure	255
23. Working with Microsoft Excel.	259
The Excel Automation Object Model	259
Converting VBA to VBScript	264
Performance Enhancements.	270
24. Working with Microsoft Outlook	275
Outlook Object model	275
Launching the Outlook Application.	276
Logging in to the Namespace	276
Enumerating the top level folder	276
Getting a Folder interactively from a user.	278
Getting the default folders	278
Reading e-mail Messages	279
Accessing e-mail Message Properties	280
Downloading attachments	281
Sending an E-mail Message.	281
Outlook Security Dialogs.	282
Getting addresses from e-mail messages	285
Extended Email Address	285
Clicking links inside an e-mail message	286
Launching QTP using an email trigger	287
25. Working with Microsoft Word	291
Word Automation Object Model	291
26. Working with Microsoft Internet Explorer.	301
Launching Internet Explorer (IE).	301
IE COM Methods and Properties.	302
Page Synchronization.	303
Enumerating all IE Windows.	304
Finding an IE window	305
Launching a new browser	305
Getting Web page DOM	307
Accessing webpage script variables.	307
Using IE to get user input	308

IE Popup Dialogs	310
Disabling IE dialogs.	310
Changing IE settings using the Registry	310
Popup Blocker	311
Disable Script Error dialog	311
Security Alert – Redirection popup	312
Security Alert – Certificate warning	313
Security Alert – Secure Connection	313
Security Information – Secure and non-secure item	314
Active content/Java Script Prompt.	315
File download – Information bar	315
Handling popup dialog by using a code	316
File Download – Security Warning popup.	317
Checking for Broken images on a Web Page.	320
Using a Browser Object as a Window Object	320
Custom Browser Applications	322
27. Working with HP Quality Center.	325
Quality Center	325
Connecting QTP with QC	325
QC Paths	327
Relative Paths in QC	327
Resolving Relative Paths	327
QCUtil Object	328
QC Open Test Architecture (OTA)	330
The TDConnection Object.	330
The Command and Recordset Objects.	333
The AttachmentFactory Collection	334
Simple way of Downloading files from QC	336
Uploading Attachments to QC.	337
Getting the Current Test Location	338
Enumerating All Tests present in a TestLab folder	338
Enumerating all the Tests in a Test Lab Tab	340
Getting the Current Test Set Location	341
28. Advanced QTP	343
Synchronizing Script Execution between different machines	343
Enumerating the Setting Variables.	344
Setting Persistence	349
Stop and Re-Run QTP using Code	350
Adding Recovery Scenarios at Run-time.	351
Executing code when Script ends	352
Making an Object visible on a Web Page	354

Advanced Text Checkpoints	354
Extending Test Objects using Classes	361
Using JScript in QTP	363
Passing variable number of arguments to a function	364
Scope differences	364
Using Try...Catch block	365
Working with JScript Arrays	366
Working with JS classes	366
Object Class Mapping	367
29. QTP 11 (New)	371
IDE Enhancement	371
Comments	371
Improved IntelliSense	371
Run to Step, Debug from Step and Run from Step	372
Enable/Disable Breakpoints	372
Renaming CheckPoints	372
Function Definition Generator	373
Library Support Enhancements	374
Web Drag and Drop Support	374
Middle Click and Right Click methods	374
Object Property Support for Firefox Browser	374
Recognizing Web objects using XPATH or CSS	375
Regular Expression Evaluator	375
Executing JavaScript Inside WebPages	376
Menu updates	377
New Configuration Options	378
Ignore Browser	378
Movie Capture to Results	379
Added and Enhanced Utility Objects	379
QCUtl	379
SystemUtil	380
The Print Log Utility	380
Hiding the Print Window	381
Showing the Print window	382
Clearing the print window	382
Programmatically Capturing the Print Log text	383
Loading and Running action at run-time	383
Repository Utility Object	383
Local System Monitoring	384
Reporting Enhancements	385
Reporting Images	385
Notes in Test Results Summary	386

MercuryTimer	386
RepositoriesCollection	387
DotNetFactory	388
Getting Object name from a TestObject in OR	388
Object Repository Enhancements	390
Local Object Repository (LOR)	390
Shared Object Repository (SOR)	392
Associating a SOR to the Test	392
Adding objects to SOR	393
Adding object using Add Object button	393
Defining the Object Manually	394
Adding object through Spy	394
Navigate and Learn	396
Other Object Related features of OR	398
Locating the object in OR	399
Updating an Object from the Application	399
Object Precedence when using both LOR and SOR	399
Object Repository Parameters (ORP)	401
Importing and Exporting OR from XML	404
Update from Local Object Repository	405
Resolving Conflicts during the Merge	407
Object Repository Merge tool	408
Object Repository Comparison Tool	409
Local OR Vs Shared OR	411
Object Spy	411
30. Working with .NET Classes	415
Using the DotNetFactory Object	415
Passing Parameters to Class Constructors	415
Passing Parameters to .NET Objects	416
Passing Enums	417
Working with .NET Arrays	418
Playing a Wav File	419
Working with the Clipboard	419
Getting Computer Information	419
Accessing the Registry	420
Ping an IP Address	420
Evaluating Keyboard Control Key Status	420
.Net Arrays Revisited	421
Working with .Net Stacks	421
Working with .Net Queues	422
Working with .Net Date and Time Formatting	422
Sending Emails using .NET	423

Converting Images to other File Formats	424
Getting user Input using .NET Forms	426
31. Designing Frameworks.....	433
Design guidelines.....	433
Support of different application versions.....	434
QTP Examples	436
Reusability	436
Support for different application versions	437
Externally Configurable.....	438
Logging	438
Self configurable	439
32. Useful Tools.....	443
QTP Script Editor.....	443
VBSEdit.....	444
Notepad++	444
QTP Uninstaller.....	445
ScreenCapture API.....	446
KnowledgeInbox IE Session Cookies Helper	447
VB 2 QTP API Converter	447
IE WebDeveloper.....	447
Test Design Studio (TDS)	449
RAD Regular Expression Designer.....	450
Appendix A.....	A-I
Appendix B: Index.....	B-I

Preface

I started with Web testing Automation in mid 2004 by creating a project in Visual Basic 6 using the Internet explorer COM Automation. The project was a great success but with every change to the application the Automation Code had to be updated and recompiled. While trying to find a solution to this maintenance issue, I stumbled upon QuickTest Professional 8.0 (QTP).

QuickTest Professional is a Test Automation tool and uses VBScript as its scripting language. QTP is a record and playback tool which can record events we perform on an application and replay them back. QTP is an object based tool which recognizes each element of the application as an object and provides various methods to work on them. All this makes QTP look like an easy-to-use test tool. The myth about Record & Playback is that it makes people think that they do not need development skills for QTP, but to create effective Automation Frameworks one needs to view QTP as a development tool and not as a testing tool. This book presents QTP as a development tool rather than a mere test tool.

One of my problems while evaluating the tool led to me to join www.SQAForum.com, without knowing that I will specialize in the use of this tool in future. After sometime I launched KnowledgeInbox.com for sharing my articles on QTP with the larger group. Dealing with day-to-day automation problems faced by people on the QTP forums, I tried solving those problems for them and learnt a few new things on my own. Observing the patterns of queries being asked on the QTP forums, I thought what the QTP community was missing is a book which can guide the amateur automation engineers in becoming a professional in the use of this tool. I took up this responsibility and started writing this book in May 2005. I spent a year on the research of the undocumented QTP features and solving the unsolved queries of QTP.

Being a first-time author, I had a very hard time getting this project completed. It was an additional responsibility, over and above my office work, QTP forum support, writing articles on KnowledgeInbox, creating tools for the community. It required a lot of motivation to keep myself focussed on the project. But knowing what difference this book can make to the QTP community always kept me motivated.

I have organized the chapters in such a way that can make learning QTP an easier task. Each chapter is based on a QTP feature. The book is divided into two sections, Basics and Advanced. Chapters 1 to 18 cover features related to QTP while the chapters in the Advanced section cover integration/interaction of QTP with various external tools like Outlook, Word, Excel and Quality Center. The book discusses a lot of issues that are commonly faced while using various features of QTP and their resolution. This book discusses almost all the topics of QTP which one would require to create complex frameworks.

Who This Book Is For

This book is for Test Engineers, Test Analysts, Test Consultants, Test Managers and anyone who is interested in learning advanced techniques of problem solving in QTP. This book is also for beginners who have just started with QTP and want to be experts in its use. The book assumes that one has the basic knowledge of QTP and VBScript, if not then it is advised that one should go through the basic help first. As the main focus of this book is to view the tool from a developer's perspective, the book does not teach how to record and replay script in QTP. Also the book does not discuss about the Keyword view of QTP, which is for non-technical people who don't want to code in QTP.

What's New in the Second Edition

QuickTest Professional Unplugged, 2nd Edition comes with all the updates till QTP 11 in a new chapter and few minor fixes in the code. The new chapter covers many of the features that got added over the previous versions of QTP. The chapter also discusses the Object Repository features in details which had seen a major change from QTP 8.X to QTP 9.

Source Code in the Book

The source codes used in the book can be downloaded from the below url

KnowledgeInbox.com/demos/QTPUnplugged_SourceCodes.zip

Feedback and Queries

For any feedback or queries you contact the author at KnowledgeInbox.com/contact-us or post a query on the KnowledgeInbox forums – Knowledgeinbox.com/forums

You can also connect with us on Facebook.



facebook.com/pages/QuickTest-Professional-Unplugged/154615194628123

Ordering this Book

For bulk orders, contact us at orders@KnowledgeInbox.com

India: You can place an order online at KnowledgeInbox.com/store

Other locations: Please check online at KnowledgeInbox.com/books

Acknowledgements

The following individuals deserve a special mention for reviewing my work in progress, and providing me with many useful comments and suggestions:

Mark Smith, Terry Horwath and Siddharth N Kanoujia

Mark works as a freelance consultant and can be contacted on Quicktest@gmail.com

Terry has not only worked as a reviewer but also as a Technical Editor for this book. Terry has worked with automated testing tools since the early 90's when he started designing solutions with Segue's QA Partner 1.0. He transitioned to Mercury Interactive tools in 1999 and has worked exclusively with QuickTest Professional since 2005. He can be contacted on thorwath@lakefolsom.com.

Siddharth has worked as an Editor for this book and has done a great job making sure that contents of this book are lucid and unambiguous for the beginners. He is an Assistant Professor in the Department of English at Hindu College, University of Delhi. He can be contacted on kandidsid@gmail.com

I would especially like to thank my family and friends who have always motivated me while I was working on this book.

Quotes From Reviewers

"I find this to be a very pragmatic, hands-on book for those who want to extend their QTP skills beyond basic expert view programming. This book is written by a QTP master for those who wish to eventually become masters themselves." – **Terry**

"Tarun Lalwani has single-handedly helped thousands of people to expand their knowledge of QuickTest Professional. Here is a book the automated testing community has been crying out for. This book will help QTP practitioners, from beginners to experts. I have used QTP from V6.0 and during the review I learnt something from every chapter." – **Mark**

"After long brainstorming sessions with Tarun over almost each and every sentence, I realized that I am truly in the presence of a genius. Tarun has with his dedication and perseverance made possible a book which will go a long way in helping people understand the ins and outs of QTP." – **Siddharth**

Chapter 1

Introduction

What is Test Automation?

Test automation is the process of reducing, and when possible, removing human interactions from an existing manual testing process.

When Should Test Automation Be Used?

Below is a list of parameters which makes a manual test a good candidate for automation:

- The test must be repeated often.
- The test's workflow and its validation evolve and change slowly over time.
- The test validates a business process or workflow, rather than *look and feel*, color, table layout, etc.
- The test is very repetitive and/or has a lot of steps, and it is important that those steps be performed exactly the same each time, where manual tester fatigue must be avoided.
- The test produces results for a regulatory body that demands that those results be electronically recorded and archived as formal evidence of compliance.
- The test's pass/fail results are reasonably easy to determine and capture with the selected automation tool.
- The test needs to drive a significant amount of data to the application

When Should Test Automation Be Avoided?

- Ad hoc testing where a subject matter expert randomly prowls through a variety of combinatorial workflows.
- One time testing or testing repeated only a few times.
- Testing which requires covering multiple functional areas such that the test travels through a small amount of virtually all of the product's functionality.
- Testing where *look and feel*, color, table layout, etc. are validated.

- Testing where pass/fail validation requires evaluating information from several different and unrelated systems and/or applications.

The Automated Testing Process

It is important to understand the various phases involved in the automated testing process in order to develop an effective framework and test cases:

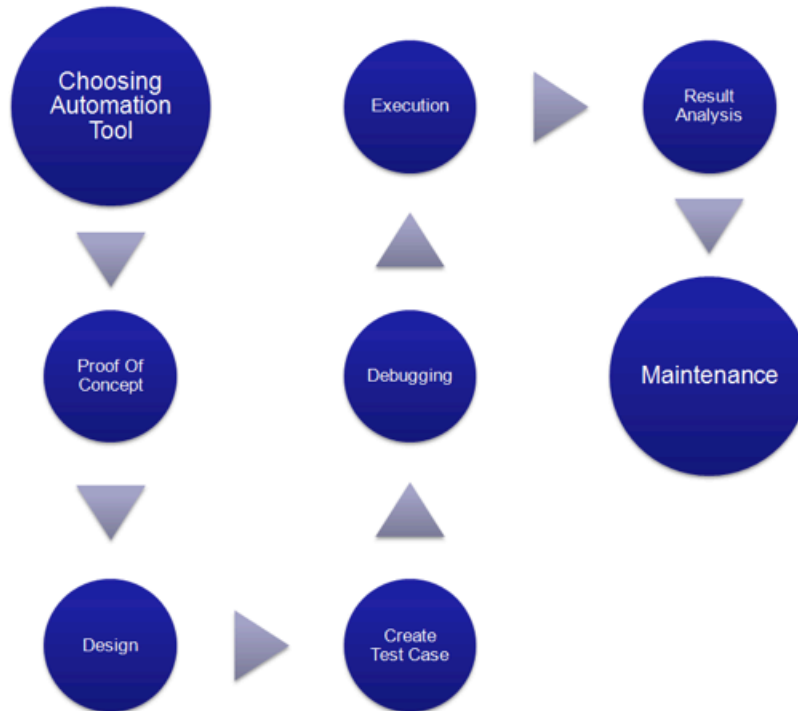


Figure 1-1. Automation Process

- Selecting the “best fit” Automation Tool:** Before starting to automate testing in any application it is important to pick the best fit tool for the majority of your applications. The choice should be based on variety of factors such as cost, ease of use, application support capabilities and product support.
- Proof of Concept (POC):** This activity involves creating a few sample scripts that validate a few important business workflows in one or two of your most important applications. This helps identify any major issues that might be encountered during future test case development. A proof of concept should also be used to select the best automation test tool for your applications.
- Requirements Analysis:** This activity involves analyzing the requirements of an application, studying the existing manual test cases and then defining the scope of the test automation project.
- Project Estimates:** Once the scope of automation is defined, estimates can then be formulated based on various factors like the number of test cases to be automated, their level of complexity, what re-usable components need to be developed, staffing requirements etc.
- Framework Design:** This activity involves creating shared object repository(s), any reusable components, writing a best practice guideline document and completing any supplementary activities to prepare a base of supporting components that will be utilized to develop automation test scripts.

- ⦿ **Test Script development:** Test cases are created by calling the reusable components and adding appropriate validations to each workflow specific script.
- ⦿ **Debugging:** Completed test cases are debugged to make sure they work as designed. Make sure you force the code through all error handling paths during phase.
- ⦿ **Execution:** In this phase test cases are finally put to work through regression testing and validating the application under test.
- ⦿ **Result Analysis:** Process followed by the results created by each executed test after execution.
- ⦿ **Maintenance:** This phase involves updating scripts to fix any code related to issues found during execution. These issues may include UI or markup changes, changes in flow or functionality or any other inevitable changes made to the application during new builds or releases. A well designed framework and set of tests ensures that maintenance costs are kept to a minimum.

What is HP QuickTest Professional (QTP)

HP QTP is a Functional Test Automation tool. Out of the box, it supports a Record and Playback methodology where automation Engineers can record and capture events with the application under test and then replay those actions during the testing phase.

QTP has two modes in which it is possible to view and edit a test script:

- ⦿ Keyword View
- ⦿ Expert View

The Keyword View displays the script in the form of keywords (each object is an icon) arranged in a tree-like format that is targeted at Subject Matter Experts (SMEs) with little or no programming background. A snapshot of the Keyword View is shown below:

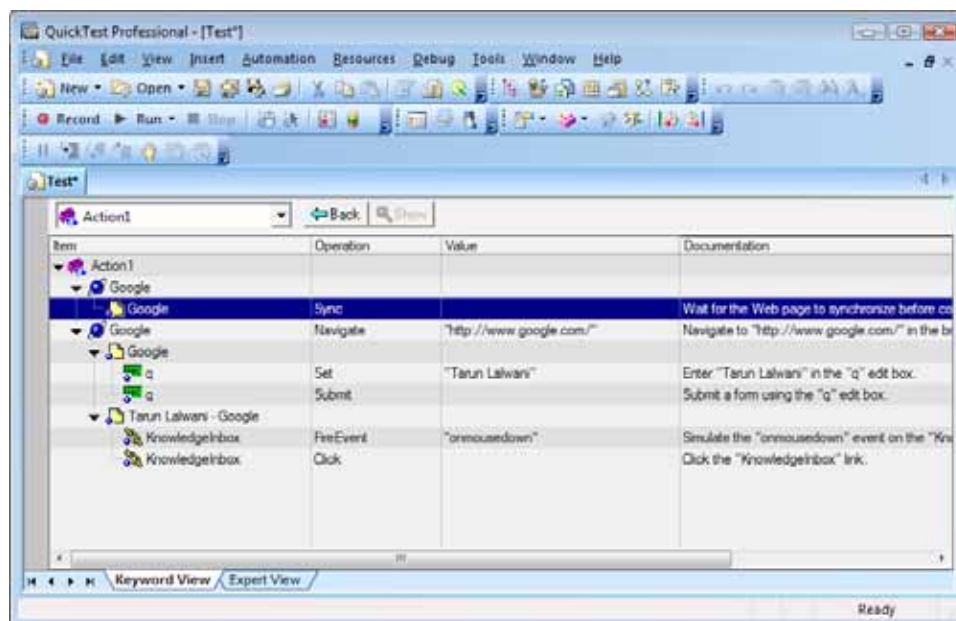


Figure 1-2. Keyword View

The Expert View exposes the underlying VBScript code shown as icons in the Keyword view and provides access to substantial capabilities not available in the Keyword view:

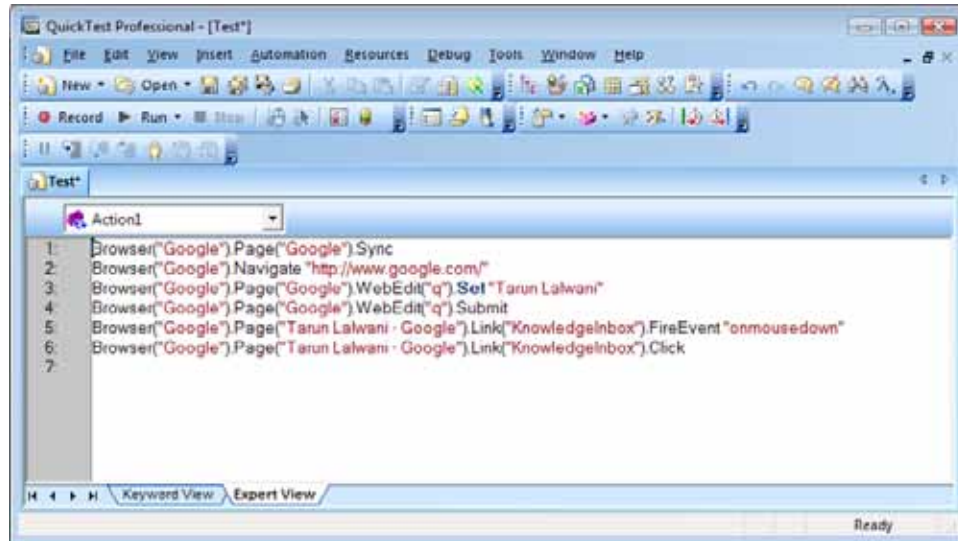


Figure 1-3. Keyword View

This book is targeted at automation engineers who want to exploit the power that QTP offers when working in the Expert View.



NOTE: QTP uses VBScript as its programming language. Virtually all VBScript functionality is available in the QTP Expert View. Conversely, any VBScript limitations are also present in QTP. It is important to learn what are QTP VBScript related in a test's structure. This book also helps clarify those aspects.

Getting Started with QTP

Downloading QTP from HP website

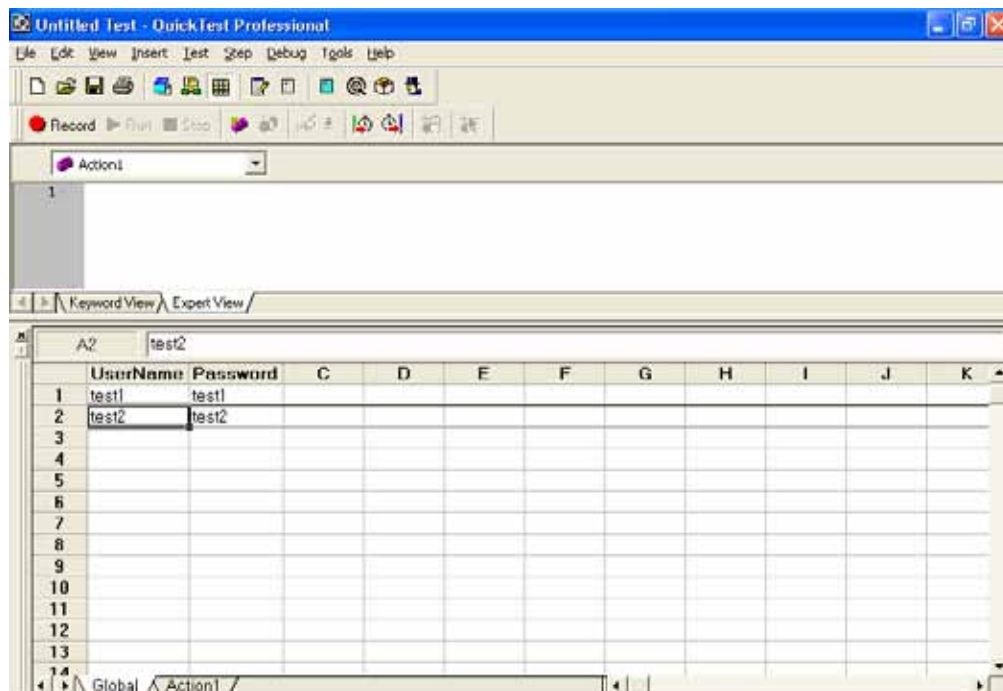
To start with QTP, first we need to download the latest version of it from the HP website. Follow the below steps to download the latest copy

- ① Go to <http://support.openview.hp.com/>
- ② Click on the Products link on the Left
- ③ Locate “HP Unified Functional Testing software” on the page and click on the link
- ④ Click on Trails and Demos and choose “HP FUNCTIONAL TESTING XX.00 EVALUATION”
- ⑤ Create a new HP passport in case you don't have one, else login using your existing passport id
- ⑥ Download the demo the file. The demo file will be in ISO format

Chapter 4

DataTables

A DataTable provides a way to create data driven test cases. It is similar to MS Excel spreadsheets and can be used to run an Action multiple times. Each test case has one global data sheet which is accessible to all actions inside that test case and each action has its own private data table also known as local data table. The name local data table is somewhat misleading because it is in fact possible to access any action's local data table from any other action, but the way of accessing the data becomes a bit different.



The screenshot shows the QuickTest Professional interface. The 'Expert View' tab is selected, displaying a DataTable. The table has columns for parameters and data rows. The first row is the header row with columns 'UserName' and 'Password'. The second row contains the values 'test1' and 'test1'. The third row contains the values 'test2' and 'test2'. The table is currently showing the 'Global' data sheet for 'Action1'.

	UserName	Password	C	D	E	F	G	H	I	J	K
1	test1	test1									
2	test2	test2									
3											
4											
5											
6											
7											
8											
9											
10											
11											
12											
13											
14											

Figure 4-1. DataTable

Figure 4-1 shows a sample DataTable with 2 parameters, Username and Password.

We can use most of the formulas that work inside a typical Excel spreadsheet. But there are some differences between a DataTable and an Excel spreadsheet. In fact a DataTable is wrapped around an Excel spreadsheet—which provides access functionality to the values but does not expose the Excel spreadsheet object model.

```
'gives the value of Parameter1 stored in
'the Global data table.
DataTable("Parameter1",dtGlobalSheet)
```

```
'gives the value of Parameter1 stored in
'the current's action local data table.
DataTable("Parameter1",dtLocalSheet)
```

The same DataTable cannot have duplicate parameter names but we can use the same name parameters in different sheets (Global DataTable and Local DataTable). Each DataTable has only 1 row enabled even when it is blank and the other rows get enabled when data is entered into a new row. A DataTable is stored as "Default.xls" file in the test folder. Figure 4-2 shows how the stored file looks like

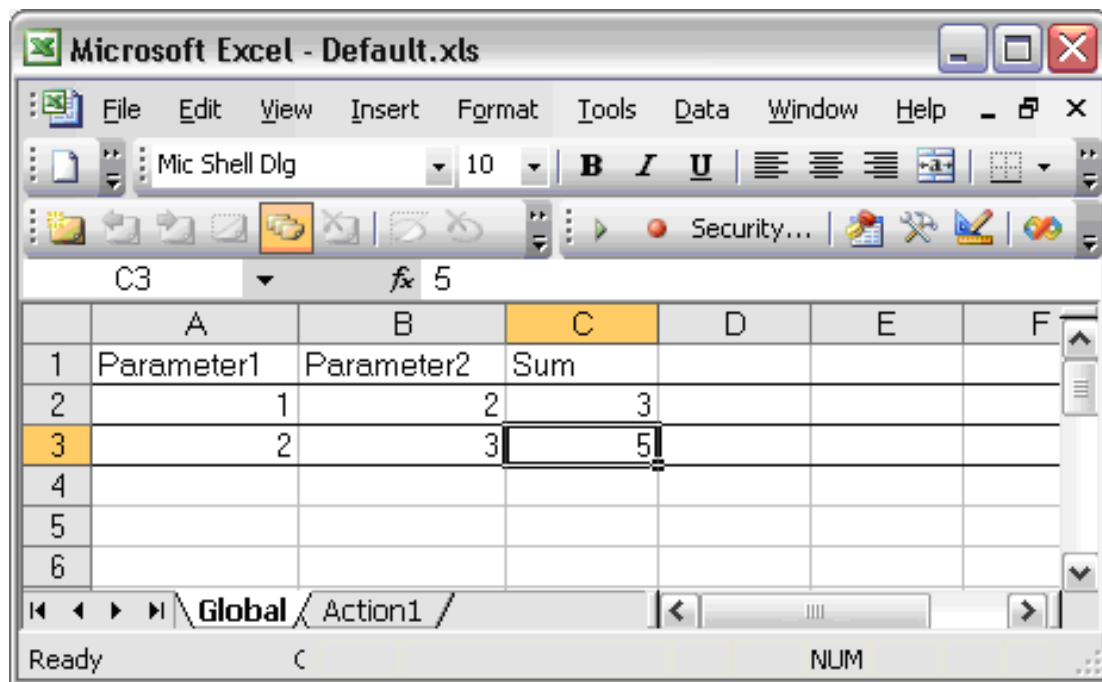


Figure 4-2. Default.xls Data table

When viewed in Excel, the first row of the sheet contains the parameter names, while QTP displays the parameter as the column titles. Therefore, when viewed using Excel, the 2nd row starts the 1st row of data in the DataTable. The DataTable shown above has only 2 data rows enabled. Note that QTP makes a data row enabled by marking the borders of the row in the actual spreadsheet. A row with no data but with marked borders is still considered as an enabled row by QTP. To delete an enabled row we must select the row and delete it from the context menu which appears on right clicking the row.

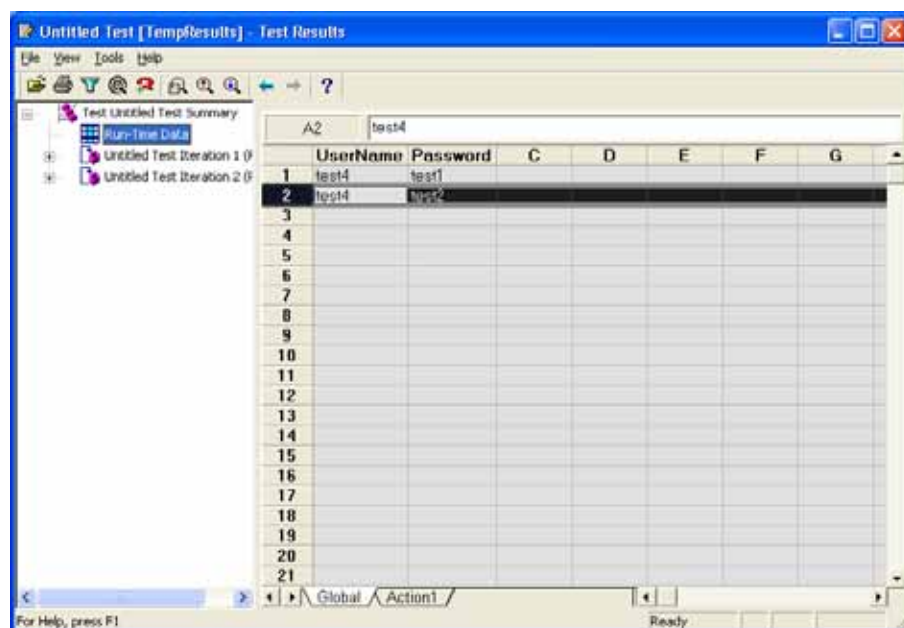
Design and run-time data table

Design time data table

As the name suggest the data table during the script design time is known as design time data table. Any changes to this are saved when the script is saved.

Run-time data table

The run-time data table contains a copy of the design time data table when a script is executed. It may contain values that are changed during script execution and are presented in the test result summary. The changes made to the data table during run-time are not saved to design time data table. Figure 4-3 shows a run-time data table from the test results summary



	A2	test4	UserName	Password	C	D	E	F	G
1	test4	test1							
2	test4	test2							
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									

Figure 4-3. Run-time data table

When to use the global or a local data table

It is important to understand in what situations the global or a local data table should be used. Consider the following two scenarios

Scenario 1 - Log into the application, book 1 ticket, log out. Repeat the scenario for many users

Scenario 2 - Log into the application, book 3 tickets, and log out

Scenario 1

The Global data table is better suited for this scenario where we have the user name, password and tickets details as the parameters and we execute the scenario using a single action (which does everything) or multiple actions (Login, booking and logout).

Data table object model

QTP provides an object model to access various properties and methods in a data table:

There are three types of objects

- ⦿ DataTable – Represents all the global and local data tables in the test
- ⦿ DTSheet – Represents a single sheet in the test
- ⦿ DTParameter – Represents a single column in a sheet.

Each object has certain functions available and certain properties associated with it. These are explained in detail in the QTP user manual.

Data table formatting

When data is entered into the data table it automatically formats the value using the best possible matching format. For example, if “12345678901” is entered into a cell then it would be auto formatted to “1.23456789E+010”. In situations where the formats are important the data should be entered with care. If data entered in the cell start with a single quote (') then it is always treated as a text and no format conversion is performed.

We can also define a specific format by right clicking the cell or an entire column and then picking a specific format from the popup context menu.

Problem 4-1. How to access a parameter from the global data sheet

There are a variety of ways to access a parameter from the global data table, most of which are presented in the following code snippet:

```
'Methods of getting a Data Table value
val = DataTable.Value("ParamName",dtGlobalSheet)
val = DataTable.Value("ParamName","Global")

'By giving the sheet index starting from 1 for the global sheet
val = DataTable.Value("ParamName",1)

' Sheet name or id is a optional parameter and is assumed
' to be as for global data sheet in case not provided
val = DataTable.Value("ParamName")

' Value property is the default property of the DataTable object
' so DataTable("ParamName",dtGlobalSheet) is
' equivalent to DataTable.Value("ParamName",dtGlobalSheet)
val = DataTable("ParamName",dtGlobalSheet)
val = DataTable("ParamName")

'Using the data table object model
val = DataTable.GlobalSheet.GetParameter("ParamName").Value

'Using the data table object model
val = DataTable.GlobalSheet.GetParameter("ParamName").ValueByRow(1)
```

Problem 4-2. How to access a parameter from a Local data sheet

```
'Various methods to get data table value
val = DataTable.Value("ParamName",dtLocalSheet)
val = DataTable.Value("ParamName","<LocalActionName>")
val = DataTable("ParamName",dtLocalSheet)
Val = DataTable("ParamName", "<LocalActionName>")

'The local sheet of the action which is executing this statement
val = DataTable.LocalSheet.GetParameter("ParamName").value
```

Problem 4-3. How to check if a Sheet exists

```
'Function to check if DataTable sheet exists
Function isSheetExists(sheetName)
    On error resume next
    isSheetExists = TRUE
    Err.clear
    Set objSheet= DataTable.GetSheet(sheetName)
    'In case error occurred sheet does not exist
    If err.number<>0 then
        isSheetExists = FALSE
    End if
End Function
```

Problem 4-4. How to preserve format of data output to a data table

```
'This would be modified to 1.23456789E+010 due to auto formatting
DataTable("ParamName") = "12345678901"
'This will not be auto formatted and will be treated as text
DataTable("ParamName") = "'" & "12345678901"
```

Problem 4-5. How to check if a parameter exists in a specific sheet

```
'Check if a parameter exists in data table
Function isParameterExists(sheetName, paramName)
    On error resume next
    isParameterExists = TRUE
    Err.clear
    ParamTotal = DataTable.GetSheet(sheetName).GetParameter(paramName)
    'In case of error the parameter does not exist
    If err.number<>0 then
        isParameterExists = False
    End if
End Function
```

Problem 4-6. How to export contents of a WebTable to a data sheet. Let's assume that the first row of the data table contains the columns heading. We then add those as parameters of the data table:

```
'Variable declaration
Dim i,j
Dim rowCount,colCount
Dim cellText, objTable
```

Problem 6-4. How can environment variables be exported to an XML file?

There are two ways to do this. First method is to write text strings to generate XML statements and second is to use the XMLUtil object to create XML objects and then export their statements. In the next example, we use a simpler method by writing strings. Working with the XMLUtil object will be covered in a later chapter.

Here is the XML code we want to create:

```
<Environment>
  <Variable>
    <Name>FirstName</Name>
    <Value>Tarun</Value>
  </Variable>
  <Variable>
    <Name>LastName</Name>
    <Value>Lalwani</Value>
  </Variable>
</Environment>
```

And here are the QTP statements to generate that XML code in an external file:

```
'Function to get XML tags for a variable with name and value
Public Function GetVarXML(varName,varValue)
  GetVarXML ="<Variable>" & vbCrLf & _
              "<Name>" & varName & "</Name>" & vbCrLf & _
              "<Value>" & varValue & "</Value>" & vbCrLf & _
              "</Variable>" & vbCrLf
End Function

'Function to write the file to a string
Public Sub WriteStringToFile(fileName, varXML)
  Dim fso, file
  Set fso = CreateObject ("Scripting.FileSystemObject")

  'Create the file
  Set file = fso.CreateTextFile (fileName,True)

  'write the text
  file.WriteLine(varXML)
  file.close

  Set file = nothing
  Set fso = nothing
End Sub
```

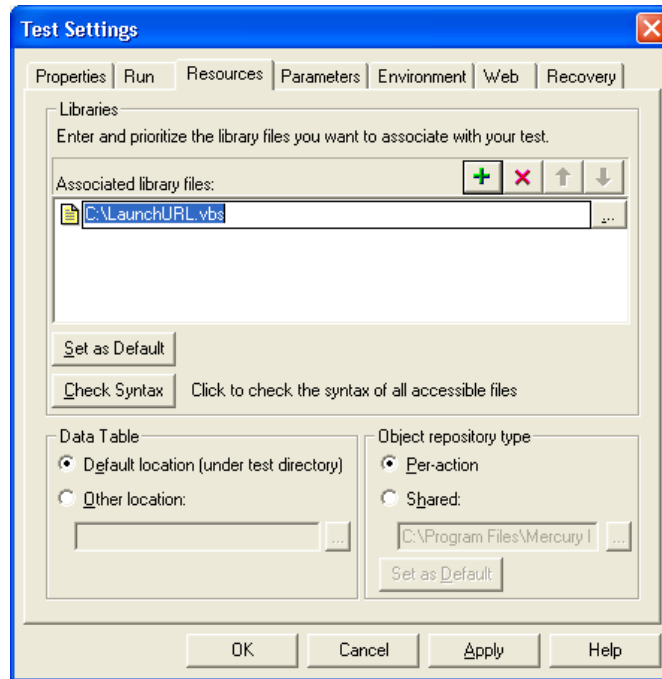


Figure 6-2. Associated library file

Now set the Record and Run setting to launch a new browser with the mercury website as shown in the Figure 6-3

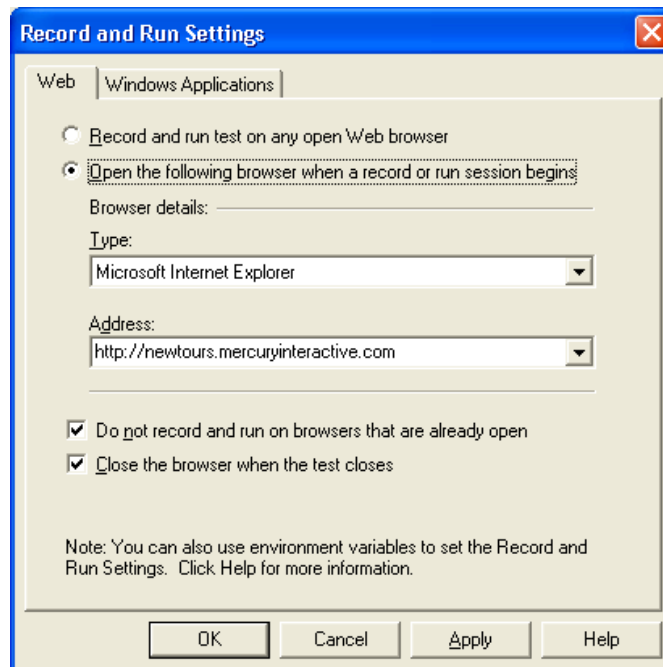


Figure 6-3. Record and Run settings

When we execute this script we will notice that the browser will open and navigate to <http://www.google.com> instead of <http://newtours.mercuryinteractive.com>.



NOTE: XML and Database checkpoints can be added in recording/design mode. All other checkpoints can be added in recording mode or through active screen objects.

Refer to the QTP user's manual for the steps needed to insert checkpoints into a test case.

Problem 8-1. How can we check if a checkpoint passed or failed?

When a checkpoint is executed its result are placed in the Test Results File. A checkpoint statement looks like below:

```
'Execute the checkpoint
Browser("").Page("").Link("test").Check CheckPoint("TestLink")
```

Here “TestLink” is the name of the checkpoint and “test” is the object on which the checkpoint is being executed. We can check the status of a checkpoint by evaluating the return value of a Checkpoint call:

```
'Save the return value of the checkpoint call
cpStatus = Browser("").Page("").Link("test").Check (CheckPoint("TestLink"))

If cpStatus Then
    MsgBox "Checkpoint Passed"
Else
    MsgBox "Checkpoint Failed"
End if
```

In cases where we don't want the status of checkpoint reported in the Test Results File, we use the Filter property of the Reporter object as follows:

```
'Disable all events reporting
Reporter.Filter = rfDisableAll

'Get the status of the checkpoint
cpStatus = Browser("").Page("").Link("test").Check (CheckPoint("TestLink"))

'Enable all reporting
Reporter.Filter = rfEnableAll
```

Problems with QTP built-in checkpoints

It's important to understand some of the problems that are encountered while using QTP checkpoints:

- ⦿ Checkpoints are not very flexible. For example, QTP does not allow the expected value of a checkpoint to be changed through the code—but we can achieve this by using DataTable variables or Environment variables to specify the checkpoint's expected values
- ⦿ Checkpoints can't be created at run-time
- ⦿ Checkpoints can't be renamed in QTP 8.x

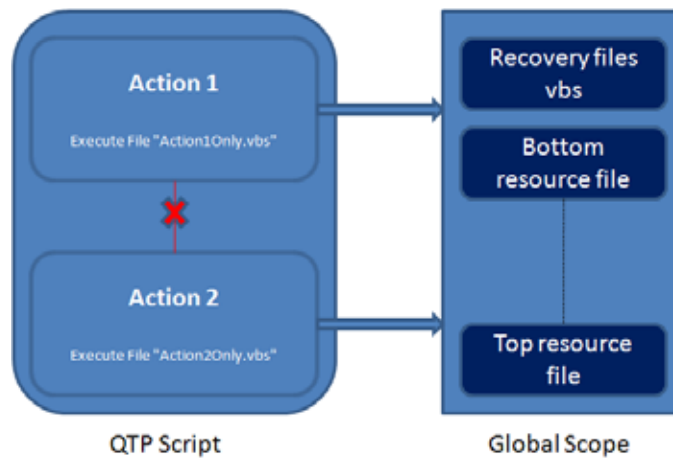


Figure 9-2. QTP Execution Scope: Local and Global scope



NOTE: If more than one library in the global scope contains a function with same name then the last one loaded would be called

Applicability of Option Explicit

Option Explicit statement at the top of a library file allows a programmer to enforce variable declaration for all used variables. We might observe that even having *Option Explicit* in some of our associated libraries is not enforcing errors for undeclared variables.

This happens because of the way Global scope is created. We cannot enforce Option Explicit on individual libraries in a global scope. It can be done only on the complete global scope. To enforce variable declaration Option Explicit should be put in every library used. If any of the libraries is missing Option Explicit then QTP will not use option explicit for the global scope.

Executing code in Local scope from within Global Scope

We earlier saw how to load library files in local scope from within the local scope (“Dynamically Loading libraries locally at run-time”) and how to load libraries in global scope from within local scope (“Dynamically Loading libraries globally at run-time”).

At times, because of script maintenance we might need to add code to the start or the end of the Action. This maintenance might apply to multiple no. of scripts. QTP allows opening single script at a time, which makes editing multiple scripts a time consuming task. To avoid such situations we can call two special functions in our Actions as shown the following code

```
'Evaluate any code that needs to be executed at
'the start of this Action
Execute GetActionStart()

'Code related to Action
```


We can now copy the above connection string into our QTP code to connect to the DB.



NOTE: In all the upcoming examples we will be assuming that we already have an adoCon object present which is connected to a DB instance

Problem 22-1. How can we execute a query to access all records?

There are various methods of doing this using ADODB.

Method 1

This method uses the Recordset object to execute the query and extract the results:

```
'Variable to get recordset result
Dim adoRecordSet

'Create the recordset. A recordset can store results
'of a database query
Set adoRecordSet = CreateObject("ADODB.Recordset")

'Set the active connection to be used for the recordset
Set adoRecordSet.ActiveConnection = adoCon

'Cursor location can take two values adUseClient = 3 or adUseServer = 2.
adoRecordSet.CursorLocation = 3 'adUseClient
adoRecordSet.CursorType = 2 'adOpenDynamic

strSQL = "Select Val1, Val2 from Table2"
adoRecordSet.Source = strSQL

'Run the SQL query
adoRecordSet.Open

'Loop till we are not at the end of the record
While not adoRecordSet.eof()

    'Access the values of the fields. We can either use the name
    'of the column in the query or use the index
    MsgBox "Val1: " & adoRecordSet.Fields("Val1").value
    MsgBox "Val2: " & adoRecordSet.Fields("Val2").value

    'Move on to the next record
    adoRecordSet.MoveNext
Wend

'Close the recordset. Recordset should always be closed
'else we won't be able to make any further queries
adoRecordSet.Close

set adoRecordSet = Nothing
```

Chapter 29

QTP 11

This chapter describes additions and enhancements added between QTP 8.2 to QTP 11. It can be important for teams to know the reason behind upgrading to QTP 11 as there has been an introduction of some important features with the latest version. This chapter can also provide a reference for teams that are unsure whether the upgrade will be a worthwhile process for them.



NOTE: Some of the features listed in this section may have been introduced in versions 9.x or 10.0. This chapter, however, provides a list of all features since QTP 8.2 to make sure your environment is using the latest version of QTP.

IDE Enhancement

QTP 11 IDE enhancements are described in this section.

Comments

You can now comment and uncomment select blocks of code. This can be done using one of the following ways:

- Using the Edit toolbar
- Using the shortcut hot keys after selecting the code block:
 - Comment Block (CTRL + M)
 - Uncomment Block (CTRL + SHIFT + M)

Improved IntelliSense

QTP 11 has improved IntelliSense as compared to its previous versions. Now, IntelliSense also works when creating tests using Descriptive Programming (which was not the case with QTP 8.x). QTP 11 also provides IntelliSense for COM objects created using CreateObject as shown in Figure 29-1.

Run to Step, Debug from Step and Run from Step

QTP 11 provides the following 3 features during debugging a test case:

- ⦿ Run to Step: This option allows QTP to execute the script upto a specific line of code and Pause where the option 'Run to Step' was selected Shortcut for this option is available using the CTRL + F10 combination
- ⦿ Debug from Step: This option starts the script in Pause mode from the specified line
- ⦿ Run from Step: This options start the script in Run mode from the specified line



NOTE: The 'Debug from Step' and 'Run from Step' options performs an event in background which can't be noticed through the QTP IDE. When we use these 2 options, QTP does nothing but removes all the code before the current line executes the scripts from that point. This is a very crude way of implementing these options. Because of this behavior, a few limitations arise. For example, it is not possible to run the script within a conditional If statement since it would generate a syntax error because of the If statement being deleted.

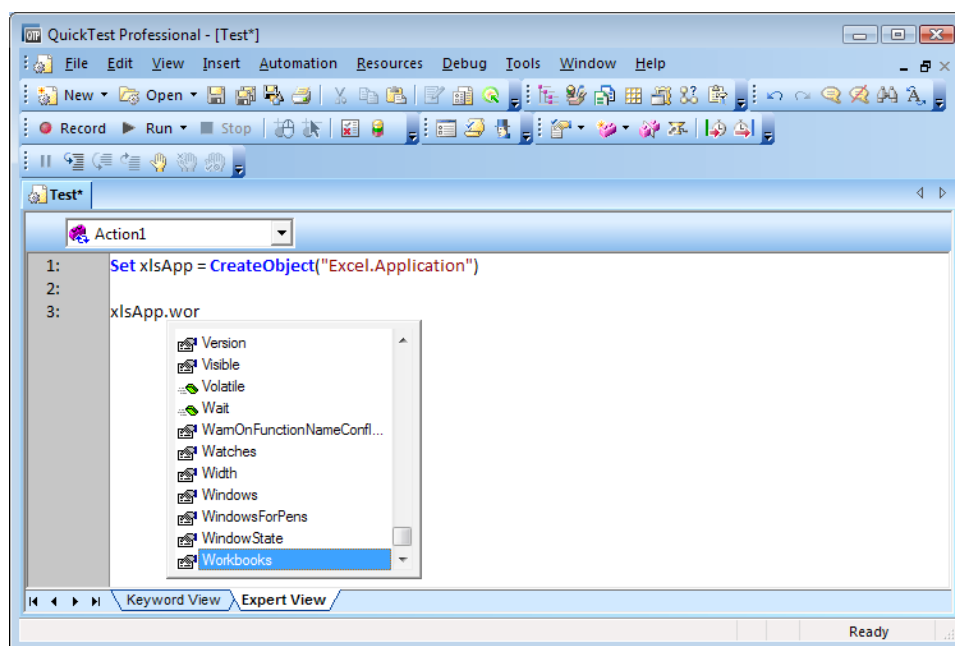


Figure 29-1. IntelliSense for COM Objects

Enable/Disable Breakpoints

QTP 11 allows you to disable and later enable existing breakpoints. In QTP 8.x you had to clear all CheckPoints to disable them but in 11 after disabling the CheckPoint the CheckPoint are still present but remain inactive until re-enabled.

Renaming CheckPoints

QTP 11 allows renaming CheckPoints which was not allowed in QTP 8.x. This feature allows teams to easily alter their code once its created without recreating the CheckPoint.

Library Support Enhancements

QTP 11 allows editing multiple libraries at the same time. It also allows debugging any library added to the Test Settings Resource tab. However, debugging of libraries loaded using the ExecuteFile statement is still not supported. To overcome this limitation, QTP 11 provides a new method called LoadFunctionLibrary to load a library at run-time while providing debugging support.

Web Drag and Drop Support

QTP 11 now supports performing drag and drop operation on web objects. This can be done by use the Drag and the Drop method on the test object.

Middle Click and Right Click methods

Most of the Web and SAP objects now support methods for Middle click and Right Click. These methods make it easier to perform operations which earlier required the use of DeviceReplay or the Click method with micRightBtn constant.



NOTE: RightClick method will fire onMouseDown, focus, onMouseUp, and onContextMenu events in series to simulate right click when the ReplayType is set as Events. To use it as proper mouse right click change the ReplayType to Mouse
 Setting.WebPackage("ReplayType") = 2

Object Property Support for Firefox Browser

QTP 11 allows accessing the Object property for Web objects in FireFox. This allows accessing the underlying DOM architecture of the web page. So the below code statement which earlier would have work on IE browser only would now work on FireFox as well:

```
'Set the WebEdit value using DOM
```

```
Browser("Google").Page("Google").WebEdit("q").Object.value = "Tarun Lalwani"
```



NOTE: IE and FireFox both use a different set of HTML DOM. Though they have few things in common but not all DOM based code written in IE will work for FireFox. For Ex – FireFox DOM doesn't have any property called outerHTML which is frequently used in IE. Though FireFox DOM does support the innerHTML property



NOTE: The properties and methods when used in Firefox are Case Sensitive. For example: nextSibling can be used as NextSibling, nextsibling or any other way in IE. However, with Firefox, it can only be used in its correct Case: nextSibling.

Recognizing Web objects using XPATH or CSS

QTP 11 also allows identifying web objects using XPATH. This makes it easier to specify few objects based on their DOM hierarchy. Instead of identifying a web object using different properties, we can just use one XPATH for the same. Consider the below HTML object

```
<INPUT class="hover" id=s size=45 value="Enter your search..." name=s>
```

Now we can identify this object using combinations of various XPATHs. Consider few of them

- ☉ `//INPUT[@class='hover']`
- ☉ `//INPUT[@id='s']`
- ☉ `//INPUT[@id='s',@class='hover']`

By using such XPATHs, we can also limit our object search. For example: if we want access to an object inside a WebTable cell, we know that the object will have a parent with a TD html tag. In that case, we can use a XPATH like

```
//TD//INPUT[@name='s']
```

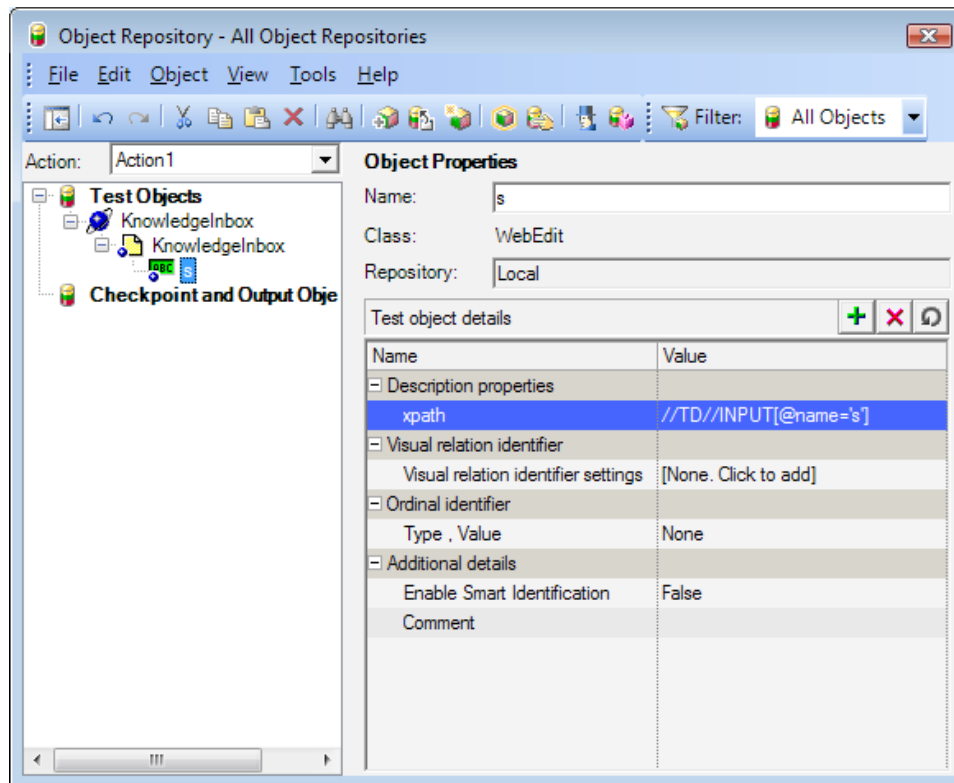


Figure 29-3. Object Identification using XPATH

Regular Expression Evaluator

QTP 11 added a regular expression testing tool. The tool can be launched from Tools→Regular Expression Evaluator.

The tool allows testing the regular expression on a sample text as shown in Figure 29-4

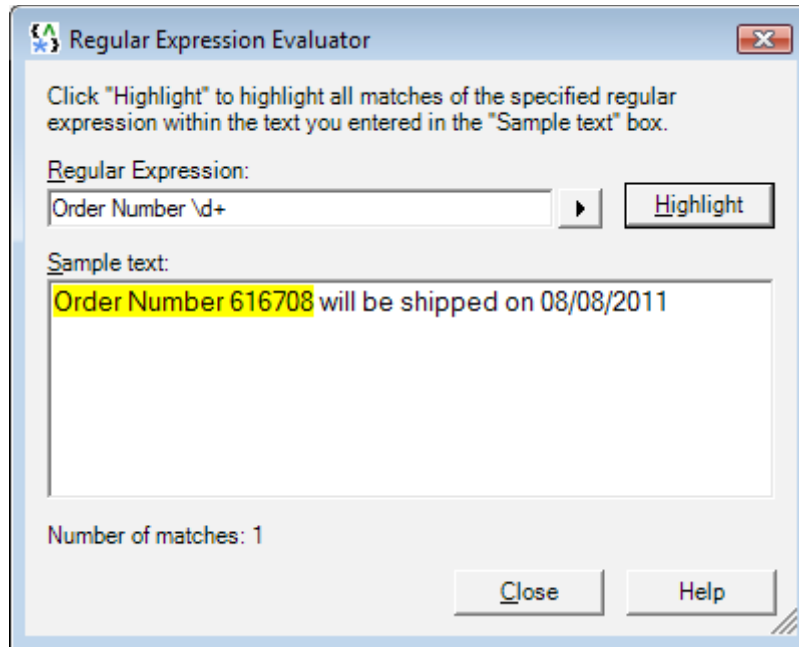


Figure 29-4. Regular Expression Evaluator

Executing JavaScript Inside WebPages

QTP 11 provides 4 methods to run JavaScript inside a web page

- Browser().EmbedScript
- Browser().EmbedScriptFromFile
- Page().RunScript
- Page().RunScriptFromFile

The EmbedScript or EmbedScriptFromFile methods can be used to embed a JavaScript in a web page. QTP will load this script every time a Page or Frame refreshes. These methods make sure the JavaScript is always available in the browser memory. These methods should be used to inject reusable JavaScript methods into the browser that might be required for testing.

The RunScript and RunScriptFromFile method provides a way to execute a JavaScript and capture its results. Consider the below code:

```
'Embed a re-usable function that we can use on any page
Browser("KnowledgeInbox").EmbedScript "function getPageTitle(){return document.title;}"

'Get the current page title
strTitle = Browser("KnowledgeInbox").Page("KnowledgeInbox").RunScript("getPageTitle()")
Print "Current page title = " & strTitle

'Navigate to a new page
Browser("KnowledgeInbox").Page("KnowledgeInbox").Link("QuickTest Professional").Click
Browser("KnowledgeInbox").Sync

'Get the current page title
strTitle = Browser("KnowledgeInbox").Page("KnowledgeInbox").RunScript("getPageTitle()")
Print "New page title = " & strTitle
```

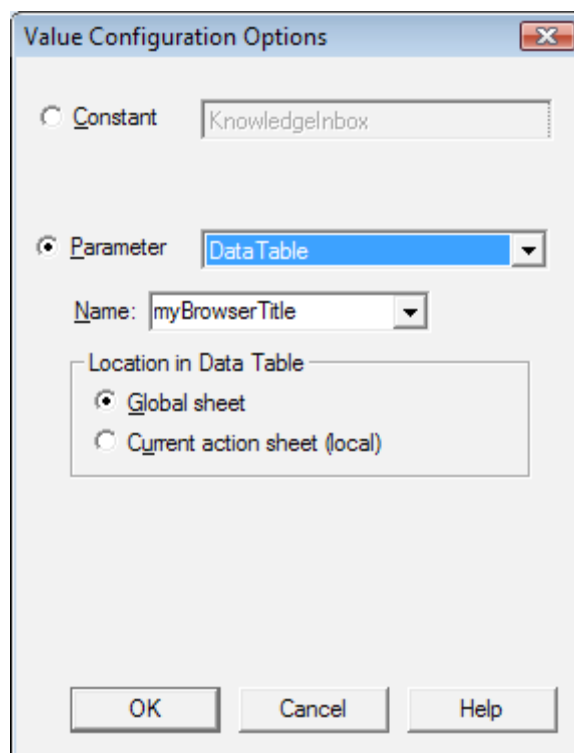


Figure 29-35. Repository Parameter value configuration



WARNING: One huge limitation of using Repository parameters is that we can't specify the value as a regular expression pattern. I am not sure how HP missed this but this feature is not available with parameterize values in OR

Importing and Exporting OR from XML

SOR can be exported as XML in a QTP defined format. The XML format is documented in QTP help file under the topic “ HP QuickTest Professional Advanced References > HP QuickTest Professional Object Repository Schema”. To Export the SOR go to File→Export to XML...

The exported SOR can be re-imported again using the File→Import from XML... option



NOTE: Exporting and re-importing the OR can at times reduce the size of the SOR and is a good way to compact the same.



NOTE: Exporting SOR to XML includes CheckPoint information only in case of QTP 11. This is not true in case of QTP 10 or lower



NOTE: Importing the XML always creates a new SOR and doesn't merge the object with existing SOR

Appendix A

Problem 3-1. Working with Test Object properties.	22
Problem 3-2. Changing Test Object properties at run time.	23
Problem 3-3. Getting Run-time Object properties during test execution.	23
Problem 4-1. How to access a parameter from the global data sheet.	38
Problem 4-2. How to access a parameter from a Local data sheet.	39
Problem 4-3. How to check if a Sheet exists.	39
Problem 4-4. How to preserve format of data output to a data table.	39
Problem 4-5. How to check if a parameter exists in a specific sheet.	39
Problem 4-6. How to export contents of a WebTable to a data sheet. Let's assume that the first row of the data table contains the columns heading. We then add those as parameters of the data table:	39
Problem 4-7. How to get value of a parameter from any specific row in the data table.	40
Problem 4-8. How to execute a script for all Global Data Table iterations, when the script is set to run for only one iteration:	41
Problem 4-9. How to get the number of columns that contain data:	41
Problem 6-1. How to access an environment variable?	53
Problem 6-2. How can we check if an environment variable exists?	53
Problem 6-3. How is an environment variable defined at run time?	53
Problem 6-4. How can environment variables be exported to an XML file?	54
Problem 6-5. How are QTP record and run settings changed using Environment variables?	55

Problem 6-6. How is an environment variable cleared or destroyed?.....	57
Problem 6-7. How an object is passed using an environment variable?.....	57
Problem 6-8. How an array is passed using Environment variables?.....	58
Problem 6-9. How are environment variables loaded from an external XML file?.....	58
Problem 8-1. How can we check if a checkpoint passed or failed?.....	70
Problem 9-1. How are global variables defined at run-time?.....	78
Problem 12-1. How can we get the status of an Action as Pass/Fail at the end of the action?	108
Problem 13-1. What will the regular expression be for a domain which can have possible value of “ test.app.com”, “qa.app.com”, “www.app.com”.....	120
Problem 13-2. What will be the r.e. matching a date in the format MM/DD/YY?.....	120
Problem 14-1. How can we find the current page and the total page values from the string “Page 4 of 15”?.....	131
Problem 14-2. How can we count the number of “if” phrases in the “if + if = 2 if” string?.....	131
Problem 14-3. Given a complete file path specification, extract the filename, file path and drive letter from the specification.....	133
Problem 14-4. Let's revisit the page number extraction problem again. How can we find the current page and the total page values from the string “Page 4 of 15”?.....	134
Problem 14-5. How can we split a string into words or lines?.....	135
Problem 14-6. How can the number of business/working days be calculated between two dates (i.e. difference should exclude the weekends)?.....	139
Problem 14-7. Get a sub-string in a string between two specified characters.....	142
Problem 14-8. Function to get a substring from a start character in a string.....	143
Problem 14-9. Write a function that reads a string word by word where two words would be separated by one or more spaces.....	144
Problem 14-10. Create a function TrimAll which trims not only spaces but other whitespace characters like new lines, carriage returns, and line feeds etc.:.....	144
Problem 14-11. Create a FilterStr function which takes a string and character set as input and will filter the string based on the characters in the character set.....	145
Problem 14-12. Create two functions, one which returns the path portion and another the file name, when passed a fully qualified path or filename:.....	146
Problem 14-13. Write a function that compares two string ignoring new line, line feeds, tab etc. whitespace characters:.....	147
Problem 14-14. Write a function that replaces repeated occurrences of a given character by	

another character.	147
Problem 14-15. How can we get the UBound/LBound values of multi-dimensional array?.....	149
Problem 14-16. How can we declare and initialize a read only variable at run-time?.	150
Problem 17-1. How can we determine if the topmost window on the desktop is a browser window?.	178
Problem 17-2. How can we get the value of a Windows Environment variable? (Note that QTP environment variables differ from Windows environment variables).....	178
Problem 17-3. How can we check (select) an item in a List box using a Windows API?....	178
Problem 17-4. How can we get the background color of a text box (helpful when a color for a mandatory field is different than an optional color)?.....	179
Problem 17-5. How can we simulate a keyboard event using the Windows APIs?.....	179
Problem 17-6. How can we prevent a PC from being locked by its screen saver?.....	181
Problem 17-7. How can we maximize a window or a browser?.....	182
Problem 17-8. How can we download a file from a URL to disk?.....	182
Problem 18-1. How can we use AOM to run a QTP test case?.....	195
Problem 18-2. How can we use AOM to change the starting and ending iterations for a test case?.....	196
Problem 18-3. How can we use AOM to launch QTP on a remote machine?.....	198
Problem 21-1. How can we copy an XML file and save it to another XML file?.....	238
Problem 21-2. How can we get an entire XML file into a string and then save it to another XML file using that string?.....	239
Problem 21-3. How can we create the following XML code at run-time in QTP, and then save it to an XML file?.....	239
Problem 21-4. How can we load environment variables from multiple XML files?.....	240
Problem 21-5. How can we modify the contents of an XML file?.....	242
Problem 21-6. How can we export a part of XML to a DataTable?.....	243
Problem 21-7. How can we extract XML from a URL or Web browser?.....	244
Problem 21-8. How can we compare two XML documents for equality?.....	244
Problem 21-9. How can we compare two xml file with different ordering of nodes?.....	244
Problem 22-1. How can we execute a query to access all records?.....	252
Problem 22-2. How can we determine the number of rows altered by an update or a delete query?.....	254

Problem 23-1. How can we invoke and terminate an instance of Excel?	260
Problem 23-2. How can we create a new excel sheet and then save it?	260
Problem 23-3. How can we determine if an Excel file exists, and if not, how to create one?	261
Problem 23-4. How can we access all non-blank row values in a specific column?	262
Problem 23-5. How can we search for a text string and then update another cell's value in that row?	262
Problem 23-6. How can we use an Excel sheet as a QTP output Data Table?	263
Problem 23-7. How can we allow a user to select a file to be opened?	264
Problem 23-8. How can we access an Excel spreadsheet embedded in Internet Explorer?	268
Problem 25-1. How to instantiate and terminate a word application?	292
Problem 25-2. How to get reference to an already open word application?	292
Problem 25-3. How to enumerate all the open word documents?	292
Problem 25-4. How to open and save a new document in word?	293
Problem 25-5. How to get reference to an already existing word document?	293
Problem 25-6. How to open a word document in the read only mode?	294
Problem 25-7. How to print an open word document?	294
Problem 25-8. How to insert text into a word document from a pre-defined style?	294
Problem 25-9. How to insert an image in a word document and scale its size by 50%?	295
Problem 25-10. How to find and replace some text in a word document?	295
Problem 25-11. How to insert a table into a word document?	296
Problem 25-12. How to change the font for the text being written in a word document?	297
Problem 25-13. How to preserve the text already present in a word document while writing?	298

Appendix B: Index

A

Actions 43
 Input and Output Parameters 43
 Insert Call to Copy... 49
 Insert Call to Existing... 48
 Insert Call to New... 47
 actionsspath 349
 AddAttachment 337
 AddParameter 40–41
 AddSheet 40
 ADODB.Command 249, 253, 255
 ADODB.Connection 249
 ADODB.Recordset 249, 252
 Advanced Text Checkpoints 354
 API using COM Objects in VB6 182
 API using Excel 185
 ApplyXSL 245
 Array 134
 Asc 137
 Automation Object Model (AOM) 195

B

Browser Vs Page Sync 159

C

CBool 137
 CByte 137
 CCur 137
 CDate 137
 Check 70
 CheckBrokenImages 320
 Checkpoints

 Alternatives to QTP checkpoints 71
 Types of checkpoints 69
 Updating Checkpoints at run-time 71
 ChildObjects 86
 Chr 138
 CInt 137
 CleanBrowserPopups 316
 CleanText 147
 CLng 137
 CloseDescendentProcesses 65
 CloseProcessByHwnd 65
 CloseProcessById 65
 CloseProcessByName 65
 CloseProcessByWndTitle 65
 Comparing XML 244
 Converting a DOM object to QTP Test Object 208
 Converting VBA to VBScript 264
 CreateDescription 85
 CreateObject 142
 CreationTime 90
 Crypt Object 61
 CSng 137
 CStr 137
 CurrentRun 329
 currentStyle 209
 CurrentTest 329
 CurrentTestSet 329
 CurrentTestSetTest 329

D

Database 249
 DataTable 33

- Data table formatting 38
 - Data table object model 38
 - Design time data table 35
 - Run-time data table 35
 - Setting data table iterations 36
 - When to use the global or a local data table 35
 - Date 138
 - DateDiff 139
 - Day 138
 - Debugging 97
 - Using Breakpoints 98
 - Working with the Command Tab 100
 - Working with the Variables Tab 99
 - Working with the Watch Expression Tab 98
 - DefaultObjRepType 349
 - DefaultTSRFilePath 349
 - Descriptive programming 83
 - Browser identification Issue 93
 - Browser Identification using a unique OpenURL Property 93
 - Browser identification using OpenTitle 93
 - Converting an OR-based script to a DP-based script 87
 - Enumerating ChildObjects 86
 - Implicit Properties 84
 - Problems with Ordinal Identifiers 90
 - Using a VBScript DP object repository 89
 - Using description objects 84
 - Using Description Strings 86
 - When to use Descriptive Programming 84
 - Working with Multiple Browser 92
 - Desktop.CaptureBitmap 170
 - DisableReplayUsingAlgorithm 348
 - Document Object 203
 - DotNetFactory
 - Accessing the Registry 420
 - Converting Images to other File Formats 424
 - Evaluating Keyboard Control Key Status 420
 - Getting Computer Information 419
 - Getting user Input using .NET Forms 426
 - .Net Arrays Revisited 421
 - Passing Enums 417
 - Passing Parameters to Class Constructors 415
 - Passing Parameters to .NET Objects 416
 - Ping an IP Address 420
 - Playing a Wav File 419
 - Sending Emails using .NET 423
 - Using the DotNetFactory Object 415
 - Working with .NET Arrays 418
 - Working with .Net Date and Time Formatting 422
 - Working with .Net Queues 422
 - Working with .Net Stacks 421
 - Working with the Clipboard 419
 - DownloadAttachments 335
 - Dynamic Arrays 148
- ## E
- Encrypt 61
 - EnumerateAllTestSets 340
 - EnumerateAllTestsinTestPlan 339
 - EnumerateSettings 347
 - Environment variables
 - BROWSER_ENV 55
 - ExternalFileName 52
 - LoadFromFile 52
 - URL_ENV 55
 - Value 52
 - Error Handling 113
 - Eval 150
 - Excel.Application 142, 187, 188, 260, 264, 353, 359
 - Excel Automation Object Model 259
 - Execute 57, 85, 149
 - ExecuteFile 76
 - ExecuteFileGlobal 77
 - ExecuteGlobal 78
 - ExecuteGlobalCode 78
 - Executing a Stored DB procedure 255
 - Executing code when Script ends 352
 - Exist 159
 - ExitAction 45
 - Exporting XML to a DataTable 243
 - Extend 362
 - extendWebTable 362
 - ExternalDataTablePath 348
 - Extern.Declare 177
 - Extern object 177
- ## F
- Filter 137
 - FilterStr 145
 - FindAttachment 334
 - FindWindow 190
 - FindWindowEx 190
 - FireEvent 160
 - FirstGlobalIteration 348
 - Fixed length arrays 148

G

GenericSet 66
 GetAddressesFromMail 285
 GetBusinessDays 140
 GetColumnName 271
 GetCurrentTestPath 338
 GetCurrentTestSetPath 341
 GetCursorPos 185
 GetDC 179
 GetDOMObjectByPath 211
 getElementsById 204
 getElementsByName 204
 getElementsByTagName 204
 GetEnvironmentVariable 178
 GetFileName 146
 GetFilePath 146
 GetForegroundWindow 177, 178
 GetIECOMByhWnd 305
 GetIECOMByURL 305
 GetIEObjectFromhWnd 268
 GetObject 142
 GetParameter 40
 GetParentOfElement 224
 GetPixel 179
 GetRandomNumberByLen 306
 GetROProperty 14, 23
 GetRowCount 41
 GetSheet 39
 GetSMTPAddress 286
 GetStrBetweenChar 143
 GetStrFromChar 143
 GetSubFolderByPath 277
 GetTOProperties 22
 GetTOProperty 23
 GetVarXML 54
 GetWindowDC 179
 GetWordFromPos 144
 GlobalIterationMode 348
 GlobalSheet 38

H

Hex 138
 Hour 138
 HowManyWeekDays 139
 HTML Button 205
 HTML Checkbox 206
 HTML ComboBox 205
 HTML Element Collection 203

HTML Element Object 203
 HTML Link 205
 HTML List Box 205
 HTML Node 202
 HTML Radio Button 206
 HTML Table 207
 HTML Text Box 205

I

IEDownloadFile 318
 InStr 130
 InStrRev 132
 Internet Explorer 301

- EnumerateIE 304
- Popup Blocker 311
- Using IE to get user input 308

 InternetExplorer.Application 302, 303, 307
 IsConnected 328
 IsEnvExist 53
 isParameterExists 39
 IsRegEqual 123

J

Join 134
 JScript 363

- Passing variable number of arguments to a function 364
- Scope differences 364
- Try...Catch block 365
- Working with JS classes 366
- Working with JScript Arrays 366

K

keybd_event 179
 KeyDown 180
 KeyPress 180
 KeyUp 180

L

LastGlobalIteration 348
 launchreport 349
 Lcase 129
 Left 130
 Len 130
 Library files 75

- Associating a Library globally with a Test 75
- Dynamically Loading libraries locally at run-time 76, 79

 LoadFromFile 58
 LocalSheet 39

LTrim 133
LTrimW 126

M

MakeObjVisible 354
MapVirtualKey 179
micclass 86
Mid 130
Minute 138
Modal dialog boxes 190
Month 138

N

NewSync 317
Now 138

O

Object Identification 27, 83
Object Spy 25
ObjRepType 348
Oct 138
On Error Resume Next 113
OnReplayError 348
Open Test Architecture (OTA) 330
Optional arguments in VBScript 150
 Using an Array of (Key, Value) pairs 152
 Using an Array of Key:=Value pairs 153
 Using an Array of values 151
 Using Null/Empty values 150
OptionalStep Object 62
Outlook.Application 276
Outlook Object model 275
Outlook Security Dialogs 282

P

PathFinder.Locate 63, 336
PathFinder Object 62
PostMessage 190

Q

QCUtl Object 328
Quality Center 325
 ResolveRelativePath 327
QuickTest.Application 174, 195, 198, 331, 343, 344, 351

R

RandomNumber Object 63
Recovery_LoginProxy 109
Recovery object 111

RegExp 123
RegisterUserFunc Object 66
RegularExpression 23
Regular Expressions 117
ReleaseDC 179
Replace 133
ReplaceMultipleByOne 147
ReplayType 162
Reporter.Filter 166
Reporter.LogEvent 172
Reporter.ReportEvent 167
Reporter.ReportPath 170
rfDisableAll 166
rfEnableAll 166
rfEnableErrorsAndWarnings 166
rfEnableErrorsOnly 166
Right 130
RTrim 133
RTrimW 125
RunAction 45

S

Scripting.Dictionary 77, 154, 155, 172, 175, 263, 304, 355, 360
Scripting.FileSystemObject 54, 167, 173, 174, 175, 246, 261, 335, 351, 353, 439
Seconds 138
SelectRadioByTextIndex 212
SendMessage 178
SetActiveWindow 190
SetCurrentRow 40, 41
SetForegroundWindow 179
SetSecure 62
Setting Object 63
SetTOPProperty 23
ShowWindow 182
SnapshotOnErrorActiveUI 348
SnapshotReportMode 169
Space 134
Split 134
SplitEx 135
StopAndReRunQTP 350
StrComp 129
String 134
StrReverse 130
Sync 157
Synchronization Points 157
SystemUtil Object 65

SystemUtil.Run 92

T

TDAPIOle80.TDConnection 332

TDConnection 328, 330

Test Arguments 197

Test Results 165

 Accessing Test Results at the end 173

 Configuring QTP to generate HTML results 170

 Converting XML Results to HTML 170

 Inserting Files in Test Results 167

 Inserting Snapshots in Test Results 168

Time 138

Timer 140

Trim 133

TrimAll 144

TrimW 126

TSRFilePath 349

TypeName 141

U

UBound 149

Ucase 129

URLDownloadToFile 182

UseExternalDataTablePath 348

Useful Tools 443

 IE WebDeveloper 447

 Internet Explorer Session Cookies 447

 Notepad++ 444

 QTP Script Editor 443

 QTP Uninstaller 445

 RAD Regular Expression Designer 450

 ScreenCapture API 446

 Test Design Studio (TDS) 449

 VB 2 QTP API Converter 447

 VBSEdit 444

User-defined Objects 28

V

ValueByRow 40

VarType 141

W

Wait 158

WaitProperty 158

WebRegSelectValue 124

Web Tables 217

 Access using OR 222

 Asynchronous Table loading 229

 Clicking inside a WebTable 227

 Exporting WebTable to a DataTable 229

 Finding a cell in a Table 231

 FindTextBySearch 232

 GetTableFromElement 225

 Getting a QTP Table from a element 230

 Getting the location of an object inside the table 225

 GetWebTableFromElement 230

 Using an object inside the table 224

 Using Index 219

 Using innerText/outerText 219

 Using Name or HTML ID 219

WebUtil Object 65

Weekday 139

WScript.Shell 174, 189, 193, 288, 301, 311, 315, 351, 439, 440

X

XMLAttribute 238

XML Attributes 238

XMLAttributesColl 238

XML Child nodes 238

XMLData 238

XMLElement 238

XMLElementsColl 238

XMLItemColl 238

XML Node values 238

XML Root node 237

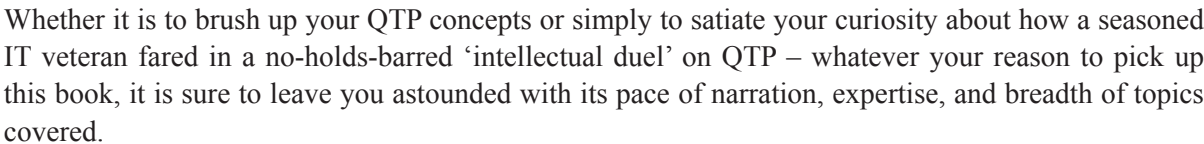
XMLUtil 238

XMLUtil.CreateXML 238

XMLUtil.CreateXMLfromFile 239

Y

Year 138



Written by the author of the best selling QTP book – “QuickTest Professional Unplugged”, this book does just that in a gripping story that will make you turn every page in anticipation. “And I Thought I knew QTP! – QTP Concepts Unplugged” is ‘different’ in the way it seeks to explain the various concepts through an interesting and innovative story-telling style (which is rarely used for technical books). Instead of following a textbook format, this book is more like a technical novel.

ISBN: 978-0-9836759-0-7



About the Author



Tarun Lalwani is a Test Automation & Solutions Architect and the author of the first ever book on QTP named “QuickTest Professional Unplugged”. He has worked on various automated testing projects utilising technologies like VBScript, VB6, VB.Net, C#.NET, Excel and Outlook Macros. He founded KnowledgeInbox.com — a blog targeted towards the QuickTest community. He uses the blog to share his custom APIs, products, articles, tips and tricks with his readers pro bono. He is also a regular contributor at AdvancedQTP and SQA forums. Tarun’s work has been showcased on several websites such as:



RelevantCodes.com



AdvancedQTP.com

Tarun was awarded with the ‘Best Feedback Award’ by HP for QTP 11 Beta testing. His book “QuickTest Professional Unplugged” was recognized as the Best Automation Book in the 2nd ATI Automation Honors award. It is the industry’s original set of awards dedicated to software test automation recognizing industry leading tools, resources and practitioners. The ATI Automation Honors awards rely on industry practitioners to identify which tools, resources and people are the best, which have the most significant upgrades and/or which are setting the trends that will help to take software test automation to the next level.

Tarun can be reached through any of the below links:



KnowledgeInbox.com/contact-us



LinkedIn.com/in/tarunlalwani



Tarun Lalwani is a Test Automation & Solutions Architect and the author of the first ever book on QTP. He has worked on various automated testing projects utilising technologies like VBScript, VB6, VB.Net, C#.NET, Excel and Outlook Macros.

He founded KnowledgeInbox.com
— a blog targeted towards the QuickTest community.

He can be reached @
KnowledgeInbox.com



QuickTest Professional Unplugged, the first book written by author Tarun Lalwani and the first ever book on QTP as well, has already turned out to be a bestseller since its publication in 2009. Tarun Lalwani has won the **Best Automation Book** award in the 2nd ATI Automation Honors for the same.

This book is good for those starting out on a career in Testing Automation or even for those with a few years of QTP experience. It is the culmination of 3 years of research and effort in this field.

The book gives a pragmatic view of using QTP in various situations and is recommended for those aspiring to be experts or advanced users of QTP.

QuickTest Professional Unplugged, 2nd Edition comes with all the updates till QTP 11 in a new chapter and few minor fixes in the code. The new chapter covers many of the features that got added over the previous versions of QTP. The chapter also discusses the Object Repository features in detail which had seen a major change from QTP 8.X to QTP 9.

Quotes from reviewers

“I find this to be a very pragmatic, hands-on book for those who want to extend their QTP skills beyond basic expert view programming. This book is written by a QTP master for those who wish to eventually become masters themselves.”

— Terry

“Tarun has single-handedly helped thousands of people to expand their knowledge of QuickTest Professional. Here is a book the automated testing community has been crying out for. This book will help QTP practitioners, from beginners to experts. I have used QTP from Version 6.0 and yet during the review I learnt something from every chapter.”

— Mark

For support and queries, please visit KnowledgeInbox.com/forums

ISBN 9780983675914

90000 >



9 780983 675914

